Stock Market Prediction using Deep Learning LSTM Model and Time Series ARIMA Model

*Samadrita Ghosh, Chis Phillips
Department of Digital Data and Technology, UK Export Finance
*samadritaghosh24@gmail.com

***Abstract*— The current living environment around us is driven by making investment and having good return on investment. People trade shares of public listed companies as buyers or sellers where they have rewarding outcome. But the nature of the existing Stock Market is very volatile and unpredictive which imposes greater risk of decision making. The computing sector is trying its best to construct a predictor model which can precisely forecast Stock Market prices under all circumstances. Main fluctuating factors in stock market prices are numerous. It can be phycological (like capitalist sentiment) or political and business related (like budgetary news) and environmental disasters (like natural calamities) and other ongoing events. All these factors contribute to increased complexity of the predictive model. Consistently experiments are undertaken and executed with a number of methods involving Machine Learning, Deep Learnings and many Time-series approaches to build an explicit predictive model. In this project, time-series model, known as ARIMA (Autoregressive integrated moving average) model and deep learning model, known as LSTM (long-short term memory) model are executed. These Deep-Learning models are implemented because they can identify paradigms and insights remarkably. The aim of the experiment is to see whether regressive or time-series model gives higher precision and accurate forecasts by comparing the prediction values and find the best fit.**

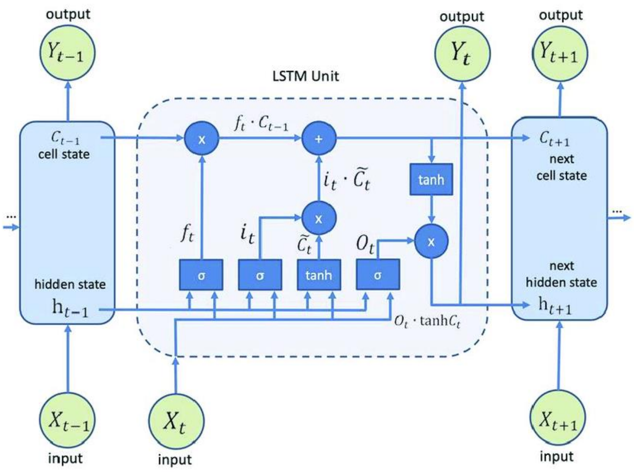
Keywords—Machine Learning, Deep Learning, Data Science, Autoregressive integrated moving average (ARIMA), LSTM (long-short term memory), Stock Market Prediction

I. Introduction

The massive upsurge of people getting involved in the Stock Markets to exchange capitals has peaked everyone's attention towards the prospects of it. Economy of our society is affected by the Stock Market prices. It has a set of defined rules with different types of input variable [1]. It becomes difficult for traditional statistics and economic models to perform well because of the complex behavior of the market [2]. The development of Artificial Intelligence, Neural Network, Depp Learning are in good shape. Different research departments have put in a lot of effort by implementing different computing algorithms and construct efficient routes of getting accurate Stock Market Prices. The achievable score by some well-known approaches like Univariate Autoregressive (AR), univariate Moving Average (MA), Simple Exponential Smoothing (SES) and different versions of ARIMA are very high and precise [2].

In this project a sequence of time-series dataset is analyzed using recurrent neural network (RNN) and auto regression (AR) method. LSTM (Long-short term memory) model follows RNN approach which is utilized to forecast Stock Market. Then the model is deployed for the end user to access it seamlessly and view the prediction. During implementation, another model known as the ARIMA model is included to check and compare the accuracy of the first model. The reason behind choosing ARIMA model is because it is very widely used and suits the nature of my dataset. Thus, while executing the project, the aim got a new direction where two models will be compared to check the best fit. It is very interesting because the models follow completely different architectures but still is able to show insistent accuracy.

In LSTM model, several recurrent neural networks (RNNs) can recognize dependencies for a longer period of time, in tasks requiring sequence prediction. [3]



$$
\begin{aligned}
\text{forget gate,} \quad f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
\text{input gate,} \quad i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
\text{output gate,} \quad o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
\text{cell state,} \quad c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
\text{hidden state,} \quad h_t &= o_t \circ \sigma_h(c_t)
\end{aligned}
$$

Figure 1: LSTM Architecture [14]

Above is the LSTM equation. Input x(t) can be the output of a CNN. h(t-1) and c(t-1) are the input from the previous steps of LSTM. o(t) is output if the LSTM for this timestep. c(t) and h(t) are generated for consumption of the nest time step LSTM [4]. The second derivative of operation is retained for a considerable length of time before the vanishing gradient needs to be addressed with zero. Thus, usage of tanh is a good approach for the earlier mentioned property [5].

ARIMA is a special kind of Autoregressive model where differencing is considered [6]. Other popular forecasting models are Multivariate ARIMA model and Vector Auto-Regression (VAR) model. They allow more than one evolving variable and derive the univariate and ARIMA model and univariate autoregressive model (AR). It will predict the data based on the past data.

II. Problem Statement

Latest futuristic aim of the project is to derive a good prediction model for Stock Market Prices of different organizations. The dataset is obtained through Yahoo Finance of 11year duration (1st January 2010 to 31st December 2021). As the data is connected directly to the URL of Yahoo Finance, the forecast of any desired organization can be checked by using the Ticker Symbol or Stock Symbol of the company. For this project, Apple stock is used with the ticker AAPL. The data is then pre-processed and run through both the models and the prediction output is fetched separately. The project started with the idea of only using LSTM Model and presenting the trend of 'actual v/s predicted' through a web application. But later ARIMA Model too was added to show the best-fit model. Conjecture is, the LSTM model will provide more accurate data than ARIMA model because it has lower error rates and learns more parameters. This feature makes of adding long-term memory makes LSTM the most powerful Recurrent Neural Network to do forecasting of longer-term trend.

III. Existing Approaches

With the advancement of modern techniques and powerful models, there are many valuable and noteworthy approaches in place with the goal to evaluate historical stock market prices and establish a good-fitting model. This will boost the dependency of the investors on accurate prediction and frequential estimation of trends. ML and deep learning prediction model reduces many unseen risk.

In the given reference [2] RNN (Recurrent Neural Network) and CNN (Convolutional Neural Network), based on Deep Learning techniques are implemented and compared to detect anomalies in time series data. RNN approach used LSTM model whereas TCN model (Temporal Convolutional Network) was implemented for CNN approach. It was noted that CNN based TCN model was portraying better performance and acting as a stable model compared to LSTM model to detect the deviation. Anomalies includes outliners, noises due to inferior quality of sensors, economical fluctuations or unforeseen changes in various factors related to stocks. Machine Learning methodologies like Support Vector mechanism (SVM), K-Nearest Neighbor and other regression is constantly being improved to deal with the anomaly changes.

An approach to verify prediction using three-layer BP Neural Network Technique is also implemented [7]. It can fit any continuous non-linear function with arbitrary precision. The approach is improved by combining random gradient decent method, but it requires huge iterations, making the approach time-consuming.

Another executable method of prediction was conducted by comparing Autoregressive Model (AR) and Functional Link Artificial Neural Network (FLANN) model which follows a Deep Learning method. It was observed that FLANN shows better performance in estimating possible forecasts as compared to KNN, BSS, MFS, MLP algorithm on stock market time- series data [8]. FLANN with back progression also exhibits minimum computation as compared to multilayer perception.

ARIMA model is the well-known time series forecasting approach. To check the performance on historical data, different Python libraries and Spark technologies are used. Prophet library was built and used upon ARIMA for non-statisticians. As a result, the model shows satisfactory prediction and sometimes better than expert projections [9,21].

Deep-learning approach via LSTM model used various stock companies and compared the prediction results based on evaluation criteria of MSE, MAE, RMSE and compare the outcomes [5].

IV. Methodology

Stock Market is very volatile and keeps changing. Thus, it is important to have an effective predictive model which can offer outstanding precision and accuracy. This project uses two discrete methods. Time-series model known as ARIMA model and LSTM is the Deep Learning model. Dataset of Apple stock price was passed through both the models to get the trendline of actual verses the predicted values. Then the mean squared error (MSE) was derived separately for both ARIMA and LSTM to indicate how accurate are the regression line and the input data points. A comparative study of the values of MSE will establish the performance of more accurate model. The model with higher precision is then attached with a webpage which runs in local host for end-users to view future predictions of any desired organization within the 11-year time-range using the Stock Ticker or Stock Symbol.

V. Implementation

Time series analysis can be executed on a server or a local laptop or desktop [9]. The implementation of this project is in Colab notebook hosted in Google's cloud servers which supports most of the libraries necessary for the implementation.

A. *Dataset*

The data repository is fetched from the Yahoo Finance website directly by connecting the source code with the URL. It captures the Apple stock data from a dynamic database. To execute this function, the data-reader library (version 10) is imported. The original dataset has 7 variables (Figure 2).

	High	Low	Open	Close	Volume	Adj Close
Date						
2012-01-03	14.732143	14.607143	14.621429	14.686786	302220800.0	12.540048
2012-01-04	14.810000	14.617143	14.642857	14.765714	260022000.0	12.607436
2012-01-05	14.948214	14.738214	14.819643	14.929643	271269600.0	12.747404
2012-01-06	15.098214	14.972143	14.991786	15.085714	318292800.0	12.880666
2012-01-09	15.276786	15.048214	15.196429	15.061786	394024400.0	12.860233

Figure 2

It includes Date, High, Low, Open, Close, Volume (Adj Close). For investors, the *close* value guides the market sentiment [10] thus, *close* column is chosen to be the target variable in this project. The data is pre-processed and split into testing data and training data in ratio 30:70 to train and test the model. Required Python packages are installed to get the libraries. The dataset is then scaled using MinMaxScaler to for optimal performance. Next stage involves feature extraction where specific features are passed into the neural network. In this case '*close*' feature was passed and rest were dropped (Figure 3).

```
test_data = df.head()
test_data['Close']

Date
2012-01-03     14.686786
2012-01-04     14.765714
2012-01-05     14.929643
2012-01-06     15.085714
2012-01-09     15.061786
Name: Close, dtype: float64
```

Figure 3 : Feature Extraction

The values of the *close* parameter is plotted in line-graph to see the trend of data. (Figure 4). It shows uptrend.

```
[<matplotlib.lines.Line2D at 0x7f53553a49d0>]
```

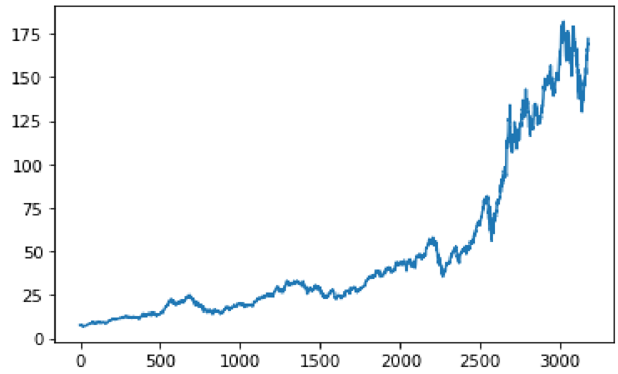


Figure 4: Trend of Close Value

.

A visual representation of the entire cycle of data is presented to understand the dataflow (Figure 5). Pre-processing of data is followed after capturing the data. Required Python packages are installed to get the libraries and facilitate easy execution of the queries. The dataset is then scaled using MinMaxScaler to for optimal performance.

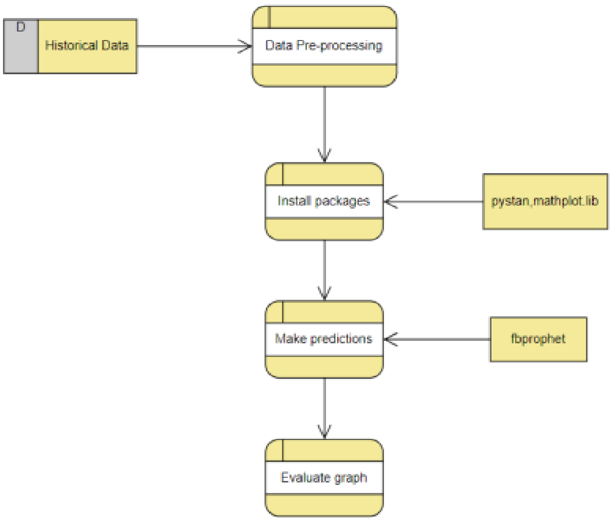


Figure 5: Dataflow Diagram [9]

After building the models, the dataset is passed through the ARIMA model first and then the LSTM model. After successful execution, the evaluated dataset is represented in graph format for quick and effortless visualization [20].

B. ARIMA Model

The objective of using ARIMA model is to get the forecast of fixed frequency time-series dataset using past values. It calculates the value by the difference between the lags of own value and the lagged forecast errors [11]. The datasets hold the characteristic of stationarity and seasonality as required for the model. The non-stationary data is converted into stationary data before execution where the past values portray the future trend.

- *ARIMA Architecture*

ARIMA is an eminent time series prediction technique. It is an integration of two models. One is based on Auto Regressive (AR) method and another is Moving Average (MA) approach [6]. Prediction accuracy of the ARIMA model can be unexpectedly low if the model is not trained with a huge dataset. [12]

The Auto Regressive property of ARIMA is referred to as P. It is a linear regression model that uses its own lags in observation as predictors [6]. It can be represented as(equation 1):

$$x_t = b + \sum_{i=1}^{p} \emptyset_i x_{t-i} + \varepsilon_t \qquad \ldots\ldots.1$$

To explain the equation, $\emptyset i$ are the autocorrelation coefficients and are calculated from delays 1 to p, x(t) is the static variable at time t, b is a constant, and the residuals are represented as εt. [13]

Moving Average is represented as Q. The degree of differencing is denoted by D. The MA is calculated as (equation 2):

$$x_t = \mu + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} \qquad \ldots\ldots.2$$

To explain, μ is the expected static variable at time t, θi the coefficients to be estimated (with $\theta(0) = 1$) . We assume that εt is a Gaussian white noise series with mean zero and variance $\sigma\varepsilon$ 2.[13]

Thus, the ARIMA model is identified as the summation of lags and lags forecast errors (equation 3):

$$x_t = b + \sum_{i=1}^{p} \emptyset_i x_{t-i} + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} \ldots\ldots.3$$

Where φi is not equal to 0 and θi is not equal to 0, and σ2 > 0. ARIMA forecasting, also known as Box and Jenkins forecasting [13]. It is capable of dealing with non-stationary time series data because of its "integrate" step. In fact, the "integrate" component involves differencing the time series to convert a non-stationary time series into a stationary. The general form of a ARIMA model is denoted as ARIMA (p, d, q). [6]

- *ARIMA Implementation*

First the necessary Python libraries -Pandas and NumPy are imported. The Apple Stock values are taken from Yahoo Finance. The data is a period of 11years. The crucial step in estimating seasonal ARIMA model is to identify the above mentioned values of (p,d,q) [6].

A screenshot of the model below shows the different features of the model: (Figure 6)

```
                               ARIMA Model Results
==============================================================================
Dep. Variable:               D.Close   No. Observations:                 2222
Model:                ARIMA(1, 1, 2)   Log Likelihood               -3054.607
Method:                      css-mle   S.D. of innovations              0.957
Date:               Fri, 12 Aug 2022   AIC                           6119.213
Time:                       01:01:36   BIC                           6147.744
Sample:                            1   HQIC                          6129.633

=================================================================================
                    coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------
const             0.0427      0.018      2.313      0.021       0.007       0.079
ar.L1.D.Close    -0.3096      0.181     -1.708      0.088      -0.665       0.046
ma.L1.D.Close     0.1654      0.181      0.912      0.362      -0.190       0.521
ma.L2.D.Close     0.0268      0.033      0.802      0.423      -0.039       0.092
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1           -3.2302           +0.0000j            3.2302            0.5000
MA.1           -3.0841           -5.2706j            6.1067           -0.3343
MA.2           -3.0841           +5.2706j            6.1067            0.3343
-----------------------------------------------------------------------------
```

Figure 6: ARIMA Model

Out of all the features, the AIC, BIC and the HQIC parameters are noted. These parameters are important to determine the performance of the possible forecast and it is elaborated in the 'results' section of the paper.

- AIC – Akaike information criteria exhibits clear skill of the model performance. It foresee and evaluate the upcoming future values based on in-sample dataset. The lesser AIC value is, the better the performance of a model. [14]

$$AIC = -2{*}\ln(L) + 2{*}k \quad \text{……..4}$$

Here L is the value of likelihood, N represents the number of recoded measurements and k denotes the number of estimated parameters.

- BIC: Bayesian Information criterion calculated the trade-off between model-fit and complexity of the model. Lowe BIC value is preferred as it indicates a better fit [14].

$$BIC = -2{*}\ln(L) + 2{*}\ln(N){*}k \quad \text{……..5}$$

The notations are same as described for the AIC formula above.

- HQC: Hannan- Quinn Criteria is a statistical criteria of model selection and is an alternative to AIC and BIC criteria. [15].

$$\mathrm{HQC} = -2L_{max} + 2k\ln(\ln(n)) \quad \text{……..6}$$

The notation L(max) is the log-likelihood, number of parameters are noted by k, n counts the number of observations. [15]

Other factors in the Python model are coefficient (coef), Standard Error (std err), Real, Imaginary and Frequency of the Auto regressive model and 2 Moving Averages.
The approach for figuring out these characteristics involves visually inspecting the time series to find trends and also looking at the correlation and partial correlation charts [11]. The flowchart of the architecture of ARIMA model is better explained in (Figure 7).

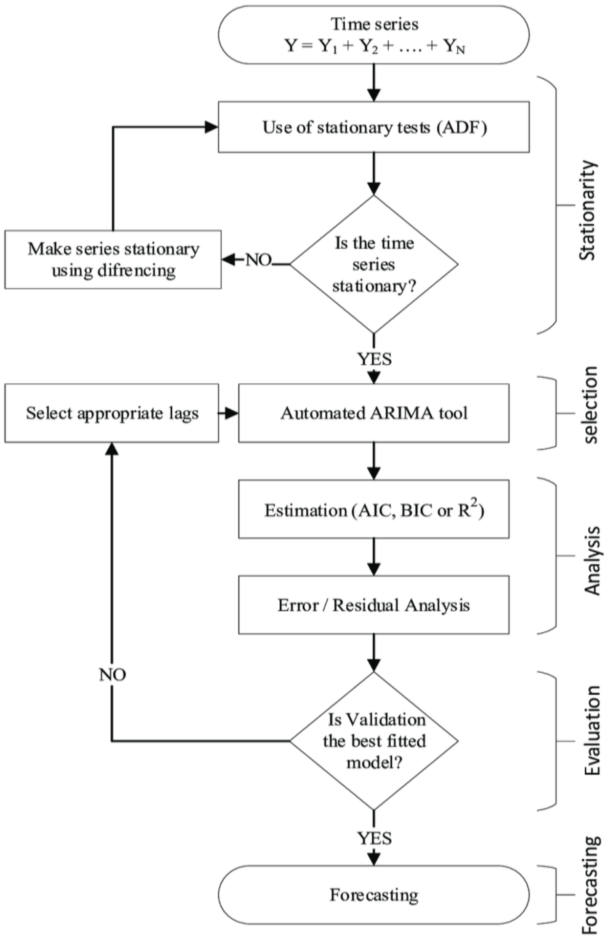


Figure 7: Flowchart to determine ARIMA Model Parameters

C. *LSTM Model*

LSTM (long short-term memory) is a recurring deep neural network (RNN) that identifies long-term dependencies and forecast a sequence of events using feedback connections and loops in the network [16]. The LSTM has linkages to feedback data and can process entire data sequence. LSTM is a unique version of RNN and exhibits exceptional performance on wide range of issues like speech recognitions, machine-translation etc. [17].

- *LSTM Architecture*

A memory cell exists in the main layer of LSTM model known as a "cell state", it stores the state over time and contributes to the key functionality in the model. 'Forget gate layer', 'input gate layer', and 'output gate layer' are the three interacting layers that make up each cell in an LSTM model [6]. The LSTM model uses these gates via each cell.

- Forget gate helps to delete data from the block.
- Input Gate contributes in the modification of memory to check if it is possible to determine which input value should be utilized.
- Output Gate determines the output by using the input and block memory. The sigmoid function determines a number that is between 0 and 1. The tanh function determines the weightage for the provided data by adjusting its significance level. [5]

These cells are built of a layer of sigmoid neural net followed by a point-wise multiplication operation [2]. The sigmoid layer outputs integers between binary 0 and 1, where 0 indicates that 'nothing should be let through' and 1 'indicates that everything should be allowed through'. A clear architecture of LSTM architecture is shown in Figure (1). Figure (8) shows the sigmoid cell:

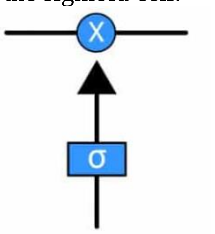


Figure 8: Sigmoid Cell of LSTM

- *LSTM Implementation*

LSTM model is used in the project because it can predict accurately in case of unpredictable delays between any occurrences in the time-series data. It can identify, process, and forecast with constrains in place. To implement the model, the Keras library imports 'sequential model' to initialize neural network, 'the density layer' to support densely connected neural network, 'dropout layer' to avoid overfitting by dropping layers and the LSTM layer. While building, four layers were added - one input layer, two LSTM layers(hidden layer) and the output layer. Every component in a layer connects to adjacent layer and the output has only single unit. The output layer also shapes the completely connected layer. In the instances of predicting stock price, three activation functions are utilized, such as: Rectified Linear Unit (ReLU), hard sigmoid and tanh. ReLU is used to enable nonlinear relationships to be learned by the neural network and is available in the output layer [5]. The flowchart of the Prediction Model Flowchart is (Figure:9)

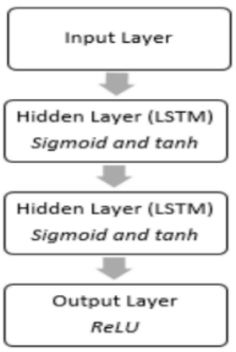


Figure: 9 Prediction Model Flowchart

The implementation of LSTM model is successfully done using python (Figure:10).

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_4 (LSTM)               (None, 100, 50)           10400

 dropout_4 (Dropout)         (None, 100, 50)           0

 lstm_5 (LSTM)               (None, 100, 60)           26640

 dropout_5 (Dropout)         (None, 100, 60)           0

 lstm_6 (LSTM)               (None, 100, 80)           45120

 dropout_6 (Dropout)         (None, 100, 80)           0

 lstm_7 (LSTM)               (None, 120)               96480

 dropout_7 (Dropout)         (None, 120)               0

 dense_1 (Dense)             (None, 1)                 121

=================================================================
Total params: 178,761
Trainable params: 178,761
Non-trainable params: 0
_________________________________________________________________
```

Figure10: LSTM Model Implementation

In the above model, *Layer(type)* is a layer array containing LSTM layer and Dropout layer. The dropout layer checks and avoid overfitting. The *Output Shape* is a 3D array (None, 100, 50) where None is the batch size, 100 is the dimensionality of the output space. Indexing this dimension, all the hidden steps can be examined at a specific time-step. For every sample in the batch there are tensor for 100 real numbers. The third dimension is set to 50 in Keras LSTM model, which denotes the dimensionality of the output space. The last column, Param #, shows the number of parameters that has undergone training in each layer. Even though all the layer of the model is trainable, it is still separated into trainable and untrainable parameter. It is mentioned at the end of the table as *Total params, Trainable params, non-trainable params.* There are no non-trainable parameters present in the LSTM model. Epoch (iterations) is run multiple time to improve the model.

VI. Analysis and Result

To perform the analysis of the performances of ARIMA model and the LSTM model, some features and parameters are considered. First, a 150-day moving average is represented in a graph along with the original data to show the price trends. Close data in blue and MA in red. (Figure 11).

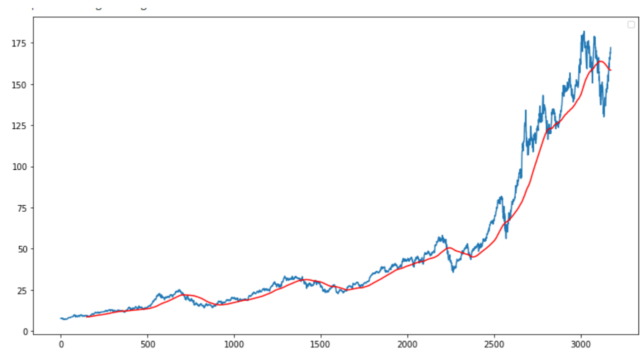


Figure 11: MA of predicted data and actual data

Higher MA makes the investors in the market very optimistic and lower MA spreads a belief that the stock market or the value of an asset will decrease. However, in practice the stock market falls soon after a high Moving Average value and rises after a significant drop. The graph (Figure 11) also shows continuous elevation or slip with no plateaued region. Since the MA fits appropriately with shift of the actual price, it can be safely said that the investors should follow and consider the trend of moving average before investing.
Moving average can be changed and set according to the investors need. In this project, a total of 4 more moving average days are considered and compared. Short term MA of 50 days and150days, long term MA of 200 days, 250days MA(Figure 12).

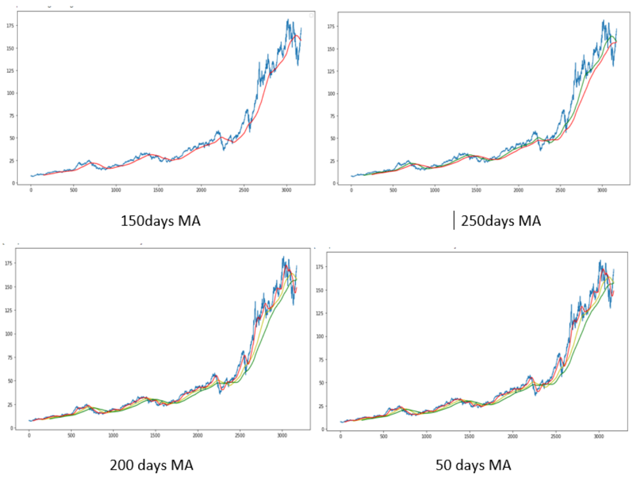


Figure 12: Comparative study of the different MA

The comparison of line-graph shows that the moving average should be considered either 50 or 200 days as short term and long term respectively.

The time-series dataset is used in the ARIMA model to check the forecasting by comparing the original price with the predicted price (Fig 13)

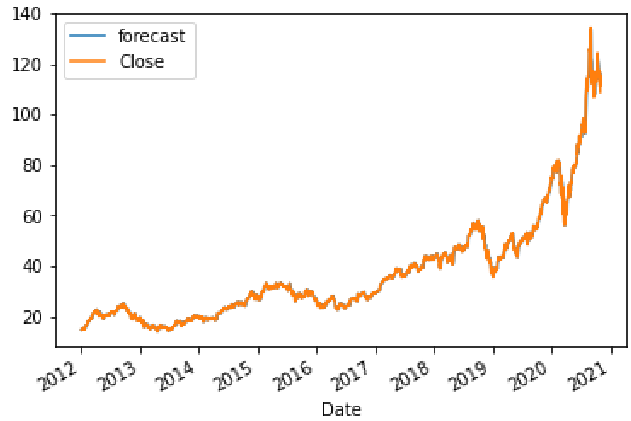


Figure 13: Prediction of ARIMA model
It is the best fit solution as the forecast data with legend blue coincides perfectly with the original *close* value.

The LSTM model has undergone 50 epoch training to improve accuracy of the model. The model is finally stored in Keras Model and the prediction is as follows (Figure 14):

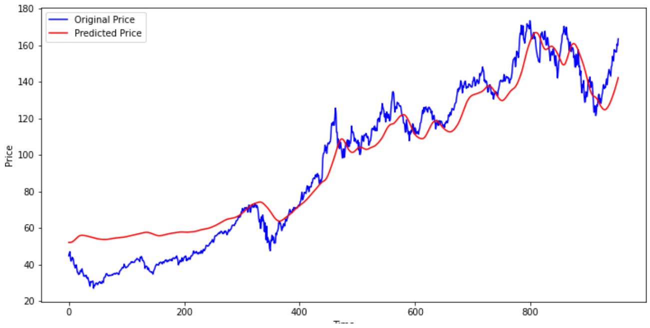


Figure 14: LSTM prediction

The prediction depicted in the graph cannot be called a best-fit as the prediction in red legend does not perfectly mimic the original Close value data in blue.

To check the accuracy of ARIMA model, residuals are extracted from the dataset. Residuals are the remaining data after fitting a model[18]. This characteristic feature of ARIMA shows the difference between the observed responses and the fitted response(fig 15).

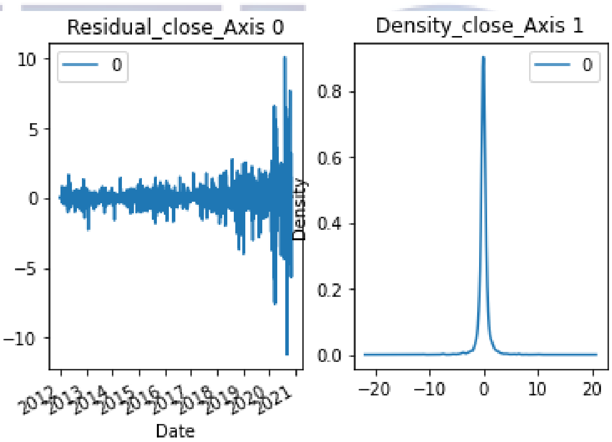


Figure 15: Residuals of ARIMA model

The density of the ARIMA model (Fig 15) shows that the highest cluster of values accumulate on gradient zero which means that the difference is negligible marking high rate of accuracy in ARIMA model prediction.

Multiple ways are there to measure and compare the accuracy of time-series models. Mean-square error (MSE) for both the models are calculated to understand the accuracy of forecasting. The average of predicted error squared values is used to determine MSE in a time-series model. The prediction error numbers are squared, which causes them to be positive and emphasizes on the divergence of accuracies [19]. There is no fixed integer to refer as ideal value of mean-square. However, a lower MSE value points to a better trained model. If the MSE value is 0.0, the model is considered to be the perfect model. The model can achieve zero MSE only when all the predicted values are equal to the expected values. This evaluation technique is most effective when the dataset contains outliers.

The MSE Value of ARIMA is calculated to be 0.0

```
{'mse': 0.0}
```

This means, the model is predicting perfectly and there is no degree of error.

For LSTM model, the data is trained repeatedly to obtain good accuracy of the model. A good MSE value must range from 0.2 to 0.5 for LSTM model. In this project, the accuracy of LSTM model is 0.0182 (Fig16).

```
Epoch 50/50
4/4 [==============================] - 1s 261ms/step - loss: 0.0182
```

Figure 16: MSE of LSTM

Based on the dataset used in the project, MSE value indicates high level of accuracy in forecasting for the ARIMA model since its close to zero. Thus, comparing the mean-squared error of ARIMA and LSTM models indicates that ARIMA can outperform LSTM for the given dataset.
Going through the implementation and analysis of this project, ARIMA model fits perfectly – this is concluded based on multiple evaluation parameters. The model high accuracy in all the comparative analysis.
In the prediction graph (Figure 13), the ARIMA model exhibits best fit as the actual and predicted lines coincides indicating zero intercept difference between the actual and predicted data.
On the other hand, the prediction graph of LSTM Model (Figure 14) also shows a good prediction potential, but deviation can be noted.
The MSE value of both models are around zero, which is a desirable score to achieve. However, ARIMA is 0.0 and LSTM is 0.018 after running through 50 epochs. Thich automatically inclines the prediction accuracy towards ARIMA model.

VII. Discussion and Future Work

The optimization process applied in LSTM deep learning approach is 'iterative' and aims at obtaining the optimal forecast-output. This makes the performance impressive because the field of error reduces and points to the most optimal score with least error [6]. Thus, the iterations enable the transformation of an inadequately fitted model into a model that is perfectly matched to the data.

The hyperparameter for iteration algorithm of training dataset are defined as Epochs. The 'Epoch' counts every instance a dataset is passed through the model to enhance the performance of it. If the Epoch displays the value as one, it means the dataset has been passed through the network only once in forward and backward directions. The number of executable Epoch is based on the complexity of the dataset used. The LSTM model runs through 50 epochs which is a feasible iteration given the complexity of data and the time required for an Epoch to run. Bigger Epochs take more time in execution. In this case, as the Epoch runs and completes each step till the limit 50, a notable decrease in mean squared error is observed (Figure 17).

```
4/4 [==============================] - 1s 271ms/step - loss: 0.0200
Epoch 43/50
4/4 [==============================] - 1s 269ms/step - loss: 0.0181
Epoch 44/50
4/4 [==============================] - 1s 269ms/step - loss: 0.0184
Epoch 45/50
4/4 [==============================] - 1s 268ms/step - loss: 0.0155
Epoch 46/50
4/4 [==============================] - 1s 264ms/step - loss: 0.0189
Epoch 47/50
4/4 [==============================] - 1s 262ms/step - loss: 0.0218
Epoch 48/50
4/4 [==============================] - 1s 269ms/step - loss: 0.0222
Epoch 49/50
4/4 [==============================] - 1s 266ms/step - loss: 0.0186
Epoch 50/50
4/4 [==============================] - 1s 261ms/step - loss: 0.0182
<keras.callbacks.History at 0x7f235f2cbd50>
```

Figure 17: Epoch run displays lower MSE with each training iteration

In future, the aim is to continue to improve the model by adding more criteria and adding more Epoch to derive a better prediction score. The most significant contribution will be connecting the backend of this source to a front-end interface where end users can utilize the findings by checking the prediction line on selecting the Stock Symbol of the company.

VIII. Conclusion

Involvement in Stock Market is fast increasing and multiple machine Learning, deep learning and other sophisticated techniques have developed many powerful models to predict values in almost all sector of research. The best-fit model depends on the nature of the data. So, we cannot comment before running the dataset through the model to find out which one will be the best-fit in a specific case. In this paper, both the models have been trained and tested using the same dataset to maintain common ground of the execution and make the research ethical.
The execution in this paper advocates how time-series analysis model is a better fit for the dynamic and seasonal financial data.

IX. References

[1] A. Sachdeva, G. Jethwani, C. Manjunath, M. Balamurugan and A. V. N. Krishna, "An Effective Time Series Analysis for Equity Market Prediction Using Deep Learning Model," 2019 International Conference on Data Science and Communication (IconDSC), 2019, pp. 1-5, doi: 10.1109/IconDSC.2019.8817035.

[2] S. Gopali, F. Abri, S. Siami-Namini and A. S. Namin, "A Comparison of TCN and LSTM Models in Detecting Anomalies in Time Series Data," 2021 IEEE International Conference on Big Data (Big Data), 2021, pp. 2415-2420, doi: 10.1109/BigData52589.2021.9671488.

[3]https://intellipaat.com/blog/what-is-lstm/#:~:text='%20LSTM%20stands%20for%20long%20short,especially%20in%20sequence%20prediction%20problems

[4] https://towardsdatascience.com/tutorial-on-lstm-a-computational-perspective-f3417442c2cd

[5] K. J, H. E, M. S. Jacob and D. R, "Stock Price Prediction Based on LSTM Deep Learning Model," 2021 International Conference on System, Computation,

Automation and Networking (ICSCAN), 2021, pp. 1-4, doi: 10.1109/ICSCAN53069.2021.9526491.

[6] S. Siami-Namini, N. Tavakoli and A. Siami Namin, "A Comparison of ARIMA and LSTM in Forecasting Time Series," 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018, pp. 1394-1401, doi: 10.1109/ICMLA.2018.00227.

[7] Y. Wang, Y. Liu, M. Wang and R. Liu, "LSTM Model Optimization on Stock Price Forecasting," 2018 17th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), 2018, pp. 173-177, doi: 10.1109/DCABES.2018.00052.

[8] S. Vijh, A. K. Pandey, G. Vijh and S. Kumar, "Stock Forecasting for Time Series Data using Convolutional Neural Network," 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2021, pp. 866-870, doi: 10.1109/Confluence51648.2021.9377128.

[9] C. R. Madhuri, M. Chinta and V. V. N. V. P. Kumar, "Stock Market Prediction for Time-series Forecasting using Prophet upon ARIMA," 2020 7th International Conference on Smart Structures and Systems (ICSSS), 2020, pp. 1-5, doi: 10.1109/ICSSS49621.2020.9202042.

[10] A. Chatterjee, H. Bhowmick and J. Sen, "Stock Price Prediction Using Time Series, Econometric, Machine Learning, and Deep Learning Models," 2021 IEEE Mysore Sub Section International Conference (MysuruCon), 2021, pp. 289-296, doi: 10.1109/MysuruCon52639.2021.9641610.

[11] M. Kesavan, J. Karthiraman, R. T. Ebenezer and S. Adhithyan, "Stock Market Prediction with Historical Time Series Data and Sentimental Analysis of Social Media Data," 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), 2020, pp. 477-482, doi: 10.1109/ICICCS48265.2020.9121121.

[12] T. Alghamdi, K. Elgazzar, M. Bayoumi, T. Sharaf and S. Shah, "Forecasting Traffic Congestion Using ARIMA Modeling," 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), 2019, pp. 1227-1232, doi: 10.1109/IWCMC.2019.8766698.

[13] I. Mpawenimana, A. Pegatoquet, V. Roy, L. Rodriguez and C. Belleudy, "A comparative study of LSTM and ARIMA for energy load prediction with enhanced data preprocessing," 2020 IEEE Sensors Applications Symposium (SAS), 2020, pp. 1-6, doi: 10.1109/SAS48726.2020.9220021.

[14] https://www.sciencedirect.com/topics/social-sciences/akaike-information-criterion

[15] https://intellipaat.com/blog/what-is-lstm/#:~:text='%20LSTM%20stands%20for%20long%20short,especially%20in%20sequence%20prediction%20problems

[16] Sepp Hochreiter, Jürgen Schmidhuber; Long Short-Term Memory. *Neural Comput* 1997; 9 (8): 1735–1780. doi: https://doi.org/10.1162/neco.1997.9.8.1735

[17] https://intellipaat.com/blog/what-is-lstm/#:~:text='%20LSTM%20stands%20for%20long%20short,especially%20in%20sequence%20prediction%20problems

[18] S. McDonald, S. Coleman, T. M. McGinnity and Y. Li, "A hybrid forecasting approach using ARIMA models and self-organising fuzzy neural networks for capital markets," The 2013 International Joint Conference on Neural Networks (IJCNN), 2013, pp. 1-7, doi: 10.1109/IJCNN.2013.6706965.

[19] https://medium.com/analytics-vidhya/significance-of-mean-squared-error-in-data-science-4c4c08da3023

[20] S. Mehrmolaei and M. R. Keyvanpour, "Time series forecasting using improved ARIMA," 2016 Artificial Intelligence and Robotics (IRANOPEN), 2016, pp. 92-97, doi: 10.1109/RIOS.2016.7529496.

[21] R. Dhoot, S. Agrawal and M. Shushil Kumar, "Implementation And Analysis Of Arima Model And Kalman Filter For Weather Forcasting in Spark Computing Environment," 2019 3rd International Conference on Computing and Communications Technologies (ICCCT), 2019, pp. 105-112, doi: 10.1109/ICCCT2.2019.8824870.